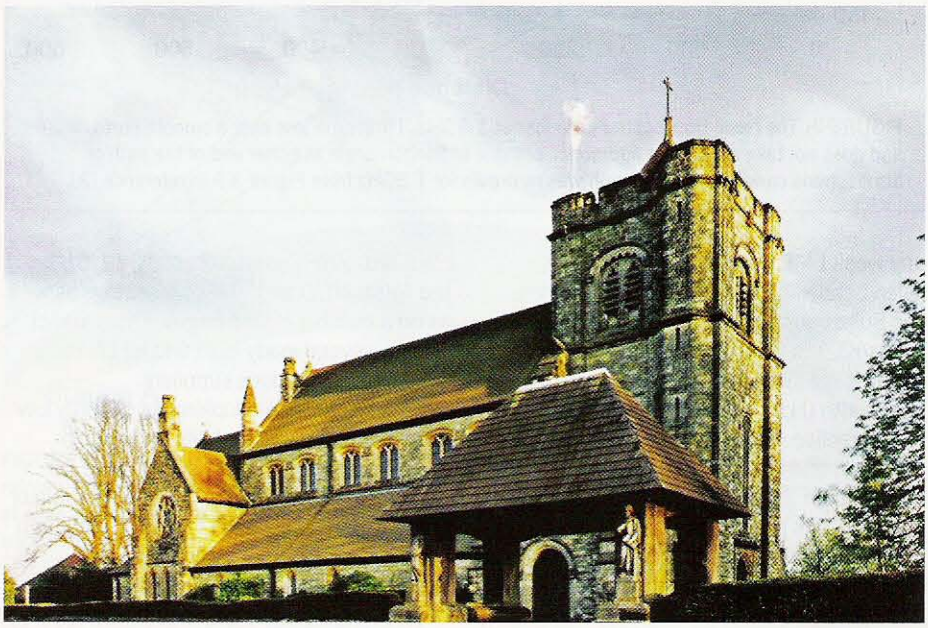
A homebrew D-Star repeater

Developing a Linux-based, G2-connected D-Star repeater on a shoestring budget



St Leonard's Church, Turners Hill, West Sussex. The flagpole atop the tower conceals the antenna for GB7MH.

IN THE BEGINNING. I became interested in D-Star during 2007/8 whilst on one of my regular visits to the USA. I'd been listening to the linked repeater networks on FM such as the excellent Winsystem, and was simply amazed by the level of traffic and the quality of coverage compared to the rather under-used repeater systems in the UK, at least round where I live.

During one of my visits, I was persuaded to pay a visit to the local Emporium and part with hard-earned money for an IC-91AD. There was a gateway-connected D-Star repeater within hitting distance of my hotel and, from that point, the benefits of linked repeaters coupled with digital audio won me over. Not having a D-Star repeater within range of my home QTH was a problem that I intended to solve; ideally without spending significant amounts of money.

As a network engineer involved in the design and deployment of voice over IP, I have many years of experience of digital voice systems. I decided, as you do, that it 'Couldn't Be That Difficult' to roll my own D-Star repeater. After all, this is just IP packets – and things don't get much simpler than IP packets do they? How wrong I was...

THE D-STAR SYSTEM. D-Star is a digital voice system that uses the AMBE-2020 codec over a GMSK carrier. There is also a low-rate user data stream carried alongside the voice signal. D-Star specifies the OTA (over-the-air) protocol, defining the all-important radio header and the format of the synchronisation carried in the data stream. A brief article like this is not the place to describe the OTA protocol in detail; the specification is open and published jointly by JARL and Icom. The AMBE-2020 codec is proprietary, owned and licensed by DVSI Inc, but this is only required if you wish to decode the voice data to baseband audio. Simply relaying the data does not require you to pull open the voice payload.

The first challenge I had was to be able to receive the RF stream carrying the D-Star signal and decode it. After many, many hours of trawling through the internet, I came across the Node Adapter project by Satoshi Yasuda, 7M3TJZ/AD6GZ.

Satoshi's board takes a GMSK input from a receiver and decodes it according to the D-Star specification. It let me see the header and the data stream through a USB interface. In transmit mode, I could send information to it and have it formatted as GMSK. Any FM rig should be able to transmit and receive the GMSK stream from the board to a commercial D-Star radio, provided the rig can modulate and demodulate down to very low frequencies.

Armed with my board from Satoshi, I went about trying to purchase the other various bits I'd need. Most were no problem – Mouser very helpfully ships to the UK, but finding the all-important CMX589 GMSK modem chip proved much harder. In the end, CML Microcircuits kindly sent me several sample chips without charge – had they not, I'd have been forced to buy from a US broker who had a minimum order charge of several hundred dollars.

I hooked the board up to a Yaesu FT-7800, plugged the USB cable into a PC and used some of the trial software provided by Satoshi. A few tweaks of audio levels later and I was sending and receiving audio! I was nearly there – or so I thought. That was July 2008.

TURNING IT INTO A REPEATER. The next step was to take the board and plug it into a system suitable for installation at a repeater site. My preference for systems such as this is to keep it simple. In a lab, a Windows system with a nice GUI is probably not going to go wrong. Translate that to the bell-tower of a church in a conservation area of southern England (the site chosen by the Ashdown Forest Repeater Group for our repeater systems), and things are going to break. It's a given. Therefore, I wanted to build the 'guts' of the system on the cheapest, least power-hungry and simplest system I could. The answer was Linux, and what I had to hand was a VIA EPIA 800 system board with 512MB of memory and a 4GB Compact Flash card – in fact identical to the system in use for my IRLP node, GB3MH. I even decided to run the same Linux operating system as GB3MH, CentOS 4.7.

So, at this point I had an RF 'hook' to the D-Star signal, a board capable of decoding it, and a system capable of 'crunching' the packets. What now?

Next came a period of understanding and learning the how D-Star systems actually work, down to bit level. As with every engineering endeavour, what I learnt was that the technical part is in fact less than 20% of the project: whilst it's a hackneyed phrase, working with other stakeholders in the various networks and solutions required diplomacy skills and collaborative working. This accounts for the other 80% of the job.

I had to learn how the current 'D-Star' network was built and identify the main characters in the system design and operation. I then had to build my relationship with them to the point where a renegade amateur (me!) could start connecting my home-brewed code into what is essentially a commercial, production-grade network.

The homebrew D-Star node lives in a professional-looking rack enclosure in the tower of St. Leonard's Church, Turners Hill.

The antenna GB7MH for is cunningly disguised as a flagpole.

SYSTEM ARCHITECTURE CONSIDERATIONS. What is thought of as 'the D-Star Network' is in fact two separate, but inter-linked systems:

- G2. Icom's network, which provides registration of users by callsign and allows callsign-to-callsign routing. Stations update their location in real time to a central Trust Server. The Trust Server is administrated by the K5TIT (Texas Interconnect Team) group, who work tirelessly to keep the network running. As a result of this effort, users can route directly to each other without having to know which system their correspondent station is on.
- D-Plus. Written and owned by Robin Cutshaw, AA4RC, this system provides hard links (much like IRLP) between repeaters and between repeaters and reflectors. Whilst it does not use G2 to route traffic, it does use the G2 database to authenticate users. It is via the D-Plus system that PC-based users can connect to repeaters and reflectors through the DV-Dongle. There is an important difference, though: while an RF user can use both G2 and D-Plus, a DV-Dongle user can only use D-Plus, ie they are not callsign routable when on their dongle.

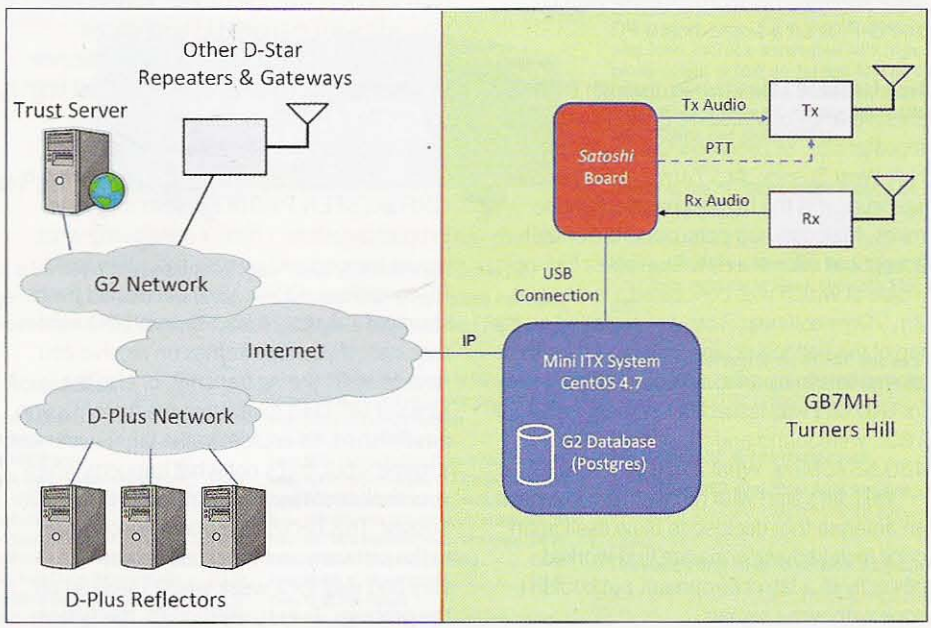
In addition, there are several other applications that run on-top of the D-Star network such as D-Star Monitor that reports last-heard information to the www.dstarusers.org page, DPRS that takes the GPS information embedded in the D-Star OTA protocol and sends it to APRS gates, etc. I had to keep all these uses in mind and make my code as transparent as possible to a D-Star user.

At roughly the same time that I started building my solution, Mark McGregor, KB9KHM, started work on his Hotspot solution. This was very useful, as we were trying to achieve similar – but subtly different – goals. Mark's goal was to build a Hotspot that provided local simplex coverage for D-Plus where there was no repeater, or where a user simply wanted their own system. His software is not linked to the G2 database, and therefore looks like a DV-Dongle. My goal was to build a replacement G2 repeater with support for the existing applications such as D-Plus. Mark and I were able to exchange ideas, which proved very useful. Before long we both had similar solutions running – Mark's on Windows, mine on Linux.

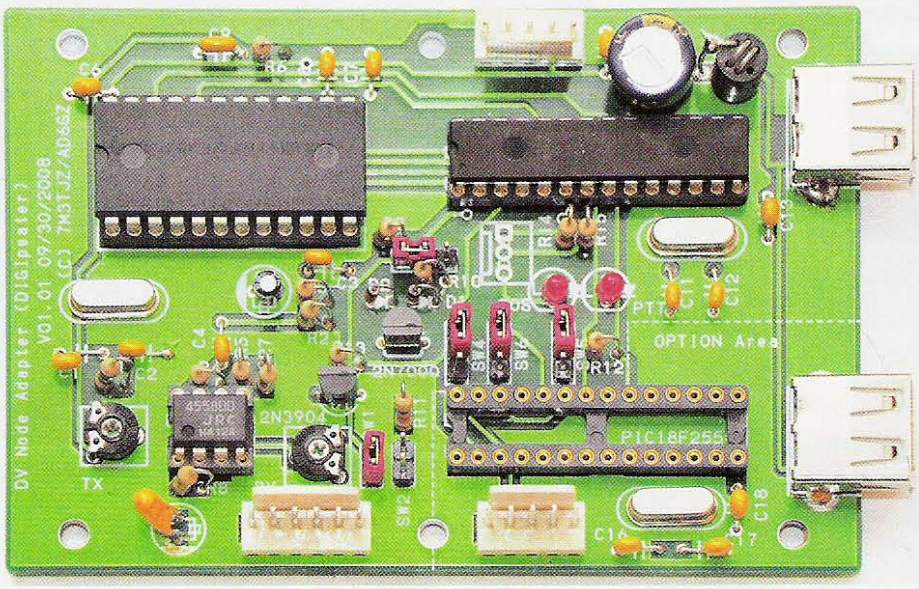
By February 2009, the first 'cut' of my repeater was running, providing local repeating services for D-Star users, and emulating a DV-Dongle to the D-Plus network. But I was not happy – this was NOT a true D-Star repeater: this was a DV-Dongle with RF. Users on my system were not G2 routable from Icom D-Star repeaters, so work carried on.

G2 COMPATIBILITY. The next stage was to start looking at the G2 gateway protocol. This is not documented, however runs unencrypted between Icom G2 systems and the Trust Server. Under the guidance of the ETCC in the UK, specifically Darren Storer, G7LWT and with immeasurable help from Jim McClellan, N5MIJ, I started to build an emulation of the Icom Gateway protocol on my system. By May 2009 I had successfully connected to the test Trust Server. With Jim's help and a test repeater system in Texas connected to the test Trust Server, we were able to hold a G2 routed QSO between Jim, mobile in Texas on his repeater, and me at home on mine!

Tentatively, my system was moved to the Production Trust Server, and GB7MH



Block diagram of major elements of the GB7MH D-Star repeater (green section) and external D-Star infrastructure.



The heart of the D-Star repeater is the Node Adapter board by Satoshi Yasuda, 7M3TJZ/AD6GZ.

appeared in the live G2 database for the first time.

Now I had to work out how to support the existing gateway applications without having to change them. I wanted to be able to take packages such as D-Plus and D-Star Monitor and load them onto my system as if they were being loaded on a real Icom G2 gateway. The answer lay in re-creating the packets that are seen in a normal Icom system.

Icom D-Star repeaters consist of an RP2C controller connecting to the radios, and a separate Linux system running the G2 Gateway code. By spoofing the packets generated by the RP2C, the D-Plus and D-Star Monitor applications on my unit thought that they could see a real RP2C and, lo and behold, they worked just as if they were on a real Icom gateway system.

I was finally there. I had a D-Star repeater that as far as users were concerned was the same as any other Icom repeater with G2 and D-Plus on a single-board PC.

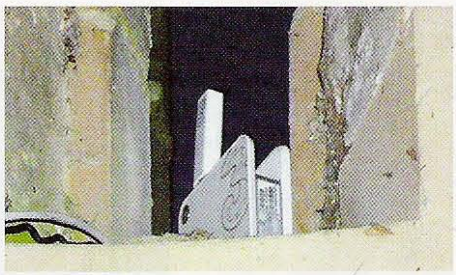
NEXT, A SITE. Meanwhile, through the sheer perseverance of Malcolm, G3NZP, we had a repeater site; St Leonard's Church, Turners Hill, West Sussex. At 174m (~600') above sea level, it is the highest point around for miles. Malcolm had persuaded the church to accept our offer of a new fibre-glass flag-pole, inside of which was concealed a radial-less 2m/70cm collinear. This was installed on the top of the bell tower, and coverage tests had proved this to be an excellent site. The NoV for GB7MH was issued, a duplexer and a Tait T800 were found and tuned for 439.6375/430.6375MHz. What could go wrong?

Well, let's start with no Internet connection, an antenna that decided to blow itself apart once installed and software that worked perfectly in a lab environment but couldn't cope with weak signals.

Getting an Internet connection to the top of a bell tower is tricky – especially one that is in a conversation area. In much of the village, no external antennas are allowed without planning permission and despite 802.11 WiFi antennas being small and discrete, none of the adjacent houses were willing to put one up to help us out. Eventually Malcolm, using his well-honed persuasive skills, convinced the church council that they'd really like a telephone in the Church Hall, and that we would pay for it. Whilst we wait for that to happen (it takes about 8 weeks for planning permission for an overhead telephone cable in a conversation area), I am using a 3G router to provide the Internet connection. It works, most of the time....

On the morning that Malcolm and I decided to install the repeater, Malcolm made one final, timely check of the VSWR of the Flagtenna. As I headed to site in the car at 08.30, I received a phone call: "I think we have a problem – the antenna is reading high VSWR across the board." Four hours later and we'd managed to wrench the antenna out of the flagpole, only to discover that the loading coil had detached itself from the radiator. Frantic repairs and another 4 hours later, the repeater was on air.

UNFORESEEN PROBLEMS. In my haste to build a system, I hadn't considered what would happen under marginal, lossy local RF conditions. D-Star uses a repeated frame structure – sync, 20 frames, sync, 20 frames, repeated. If you lose frames on receive you have to in-fill during transmit, or else the receiver will start to assume the transmission has finished. Of course, in the lab, I never lost a frame – but that's not what happens when your mobile station is driving around leafy Sussex. This necessitated a major change to the software and the repeater was in a very bad way for a week whilst I worked on the problem. Eventually though, the system was fixed and has been providing very good service ever since.



In the absence of a phone line, internet connectivity was provided via a 3G internet dongle & Wi-Fi router from 3.

My background in VoIP and Unix certainly helped – most of the coding is in C++, but aspects such as the SQL database (I use Postgresql for the G2 routing) and USB interactions (I use libusb1 and libusb-compat to access the Satoshi board) were new to me, and required many hours of testing and investigation.

IT REALLY WAS CHEAP. Compared to purchasing an Icom system for GB7MH, the hardware cost has been miniscule: about £50 for the Satoshi board and parts, less than £30 for the Mini ITX Linux system, plus a fortuitously free Tait T800 repeater and duplexer. And I've certainly learnt a lot more about how D-Star really works than had I just bought an Icom box...

After the success of GB7MH, GB7BM was created the same way and is now on-air and providing good service. GB7NE will hopefully follow soon. So the basic goal of providing a cheap but fully-compliant alternative to an Icom D-Star repeater is complete. I'm now turning my attention to improving aspects of the current D-Star network such as the extended database synchronisation times, the use of more robust VoIP protocols and new features such as Simplex G2/D-Plus nodes.

THANKS TO... And finally, none of this would have been possible without the considerable help and support of the following people: Prof Satoshi Yasuda, 7M3TJZ/AD6GZ, Darren Storer, G7LWT, Iain Philipps, G0RDI, Jim McClellan, N5MIJ, Robin Cutshaw, AA4RC, Mark McGregor, KB9KHM, John Baxter, G8VIQ, Ashdown Forest Repeater Group, Malcolm Harman, G3NZP, Richard Hadfield, G4ANN, Stewart Bryant, G3YSX and Paul Philips, G7KBR. The home page of the repeater development is http://g4ulf.blogspot.com.

WEBSEARCH

Winsystem - www.winsystm.org
Detailed D-Star protocol - www.jarl.com/d-star/shogen.pdf
DVSI (AMBE codec) www.dvsinc.com
7M3TJZ/AD6GZ node adapter project - http://d-star.dyndns.org/node_adapter.html.en
Mouser Electronics (USA) - www.mouser.com
CML Microcircuits - www.cmlmicro.com
Ashdown Forest Repeater Group - www.afrg.org.uk
K5TIT Texas Interconnect Team - www.k5tit.org
DV-Dongle - www.dvdongle.com
D-Star Users - www.dstarusers.org